第1章

プログラマブルなLSIを実現するしくみとアプリケーションを理解する

FPGAの基礎知識

三上廉司

本章では，これまでFPGAを使ったことのない人のために，FPGAの構造やFPGAの使われ方について解説する．FPGAは所望の機能を手元で構成できる特徴を持つ．これからのハードウェア開発には不可欠なLSIである．　（編集部）

最初のプログラマブル・デバイス(PLD：programmable logic device)が登場して30年，FPGA(field programmable gate array)が登場して20年がたちました．この間にプログラマブル・デバイスの高集積化，多機能化，高速化，低価格化が進んでいます．シンプルなアーキテクチャで小規模だった時代からプログラマブル・デバイスを使っている人にとって，デバイスの進歩はありがたい限りです．しかし，これからプログラマブル・デバイスを使おうというエンジニアにとっては，最新のFPGAとその設計技術は，複雑すぎて導入の糸口がつかみづらくなっているかもしれません．

ここでは，ディジタル回路の一応の知識があり，これからFPGAを活用し始めようとするエンジニアの方々を対象に，FPGAの基礎について解説します．

1．FPGAの構造を理解する

FPGAは，図1のように，プログラム可能な比較的小規模の論理ブロックを格子状に配置し，その間に縦方向と横方向に配線路を設けた構造を基本としたLSIです．一つの論理ブロックは小規模ですが，多くのブロックを組み合わせることで大規模な回路を実現できます．

図1　FPGAの基本構造
格子状に配置された論理ブロックと，論理ブロック間を接続するスイッチ・マトリックスで構成される．

● 基本論理ブロックはLUTとフリップフロップで構成

図2にFPGAの論理ブロックの基本的な構成を示します．

プログラム可能な論理ブロックを実現する基本要素は，SRAMで構成されたLUT(look-up table)です．図2では，4入力1出力のLUTの例を示しています．4ビットの入力から1ビットの出力を得る任意の論理関数を構成できます．

フリップフロップは，同期出力を得たり順序回路を構成する際に用います．LUTの出力にフリップフロップを接続するかどうかは選択可能です．

このLUTとフリップフロップのペアを基本として，高性能化するための付加回路などによって基本論理ブロックが構成されます．このブロックの構造はメーカによって異なります．CLB(cofiguarable logic block)やLE(logic

KeyWord　プログラマブル・デバイス，PLD，FPGA，SRAM，LUT，トランスファ・ゲート，IPコア
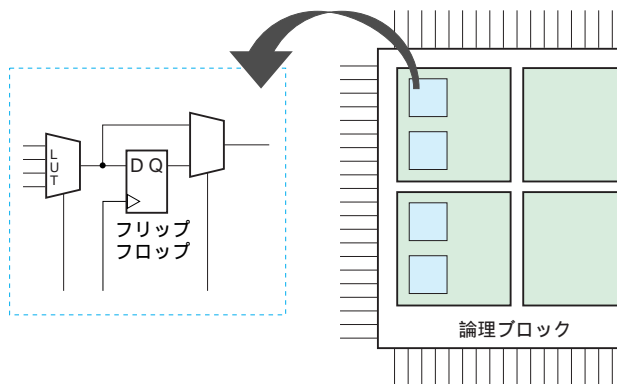
図2　FPGAの基本論理ブロック
LUTはメモリで構成される．接続はトランスファ・ゲートで作られる．

図3　トランスファ・ゲートの構造
メモリでトランスファ・ゲートのON/OFFを決める．

element)，スライスなどメーカにより名称も異なります．

● 論理ブロック間はスイッチ・マトリックスで接続

基本論理ブロックは，配線路に設けられたスイッチ・マトリックスによって任意の接続が可能です．

スイッチ・マトリックスは，図3に示すようなトランスファ・ゲート(アナログ・スイッチ)で作られています．

FPGAの論理を構成するLUTはSRAMです．スイッチ・マトリックスを構成するトランスファ・ゲートのON/OFFを決めるのもSRAMです．

SRAMは，電源がOFFになるとデータが消えてしまいます．そのため，FPGAは電源を投入すると外部から回路情報(コンフィグレーション・データと言う)を読み込みます．

● FPGAの性能は配置配線で決まる

FPGAは，同じ論理であっても，配置配線の結果によって信号の遅延時間(最大動作周波数)が変化します．多数の小規模な論理ブロックと配線リソースを組み合わせているため，論理ブロックが実際に配置される場所によって配線長や経由する配線スイッチの数が異なるためです．また，微細プロセスで製造されるLSIでは，論理素子の遅延時間よりも配置配線の遅延のほうが大きくなります．

● 1個から製造できるカスタムLSI

FPGAは標準LSIとして製造され，提供されています．しかし，プログラマブルな特徴を持ち，機能を自由に設計できるので，カスタムLSIとして活用できます．

ASIC(application specific IC；特定用途向けIC)では，ゲート・アレイやスタンダード・セルがよく使われます．これらは，半導体工場で生産するために，高い開発費と一定の製造期間がかかります．これに対し，FPGAは，回路の設計コストはかかるものの，回路データの書き込みのコストはほとんどタダのようなものです．また，設計者の手元で1個からでも製造(プログラム)できます．

2．FPGAが活躍するアプリケーション

FPGAの製造コストは，同じ回路規模で比較すると，ASICと比べて高くなります．しかしASICで必要な高額な開発費がFPGAではかからないため，数千個程度までのトータル・コストはFPGAの方が安く済むのが普通です．

● 少量多品種のアプリケーションはFPGAの独壇場

少量多品種の開発では，FPGAの独壇場と言えます．かつては高い性能が求められる場面でFPGAは使えませんでしたが，現在ではこのようなケースは少なくなりました．

医療機器分野にも画像処理が必要なMRI，超音波断層診断装置，内視鏡などの装置にFPGAがよく使われています．専門医による画像の見え方などの細かい要望やマン・マシン・インターフェースの改善などに柔軟に対応できるからです．また，工場の生産設備や冶具などにも数量が少なくてもよいというメリットが生かされています．

● 量産品へ搭載され始める

近ごろは，セットトップ・ボックスや液晶テレビといった民生機器でもFPGAが使われるようになりました．FPGA自体の低コスト化はもちろん，ハードウェアがプログラマブルに変更可能という利点があるためでしょう．

近年，製品のライフ・サイクルが短期化しています．標準とされる技術も頻繁に変化します．短期間に製品を開発

するためには，最終仕様が決まる以前に暫定仕様で開発に着手しなければなりません．開発途中に仕様変更があっても，FPGAであれば対応できます．

ASIC開発のリスク回避で活用される場合もあります．90nm以下の製造プロセスによるASIC設計では，1回の開発費が数千万円もかかります．設計完了後にバグや仕様変更が発生した場合，この費用がむだになり，さらに数千万円の費用が必要になります．実際には，バグや仕様変更などで，かなり高い確率で再設計が発生しているのが現実です．そこで比較的小規模な(つまり低価格の)FPGAを使ってハードウェア的なパッチを当てることがあります．

● 組み込みシステムへの採用が進む

汎用LSIの組み合わせで実現可能な組み込みシステムにも，FPGAが活用されるケースが増えています．

プリント基板には信号の伝送特性の制約があるため，回路の高速な動作は難しくなります．しかし，複数のLSIを1チップ化してしまえば，プリント基板上で配線を行う場合と比べて高速な動作が可能になります．汎用LSIと比べてFPGAは高価な場合がありますが，1チップ化によってLSIのコストだけでなく，実装コストの低減が期待できます．

FPGAで汎用LSIと同等の機能を実現するためには，IP(intellectual property)コアを活用することができます．

CPUのIPコアも広く活用されています．FPGAにはCPUコアを専用機能ブロックとして搭載したものと，合成可能なコアとして提供され論理ブロックで実現するものがあります．

写真1　FPGAデータも送受信している基地局のアンテナ
ウィルコムの大規模公衆ワイヤレス・ネットワークでは，基地局のハードウェア(FPGA)更新を無線伝送で行っている．

● 再構成で仕様変更に対応する

FPGAは，DSPとともに携帯電話やPHSの基地局(ベース・ステーション)にソフトウェア無線用として広く使用されています(**写真1**)．通信規格やプロトコルの変更は，FPGAの回路の再構成で対応します．

3．FPGA開発の現在と将来

FPGAをはじめとする現在のLSI設計では，設計仕様を言語で記述し，論理シミュレータで検証し，設計自動化ツールにより回路を合成/配置配線します．

FPGA開発で使われるハードウェア記述言語(HDL：hardware description language)は，Verilog HDLとVHDLの二つが主流です．

FPGA設計では当初VHDLが広く利用されていました．VHDLにしか対応していないFPGA向けの設計ツールが多かったためです．しかし最近ではVerilog HDLを使うFPGA設計者も増えているようです．最近の設計ツールは，両方の言語に対応していることと，従来ASICを設計してきた人たちがFPGAを活用し始めているためでしょう．ASIC設計では，サインオフのシミュレータの関係でVerilog HDLが広く使われています．

近年，Cなどのプログラミング言語からFPGAを設計する手法も注目を集め始めています．C言語などのアルゴリズム記述から回路を合成する技術は急速に進歩を遂げていますが，一般に普及するのはこれからという段階です．

みかみ・れんじ
ミカミ設計コンサルティング

<筆者プロフィール>
三上廉司． PLD/FPGAとは1970年代の出始めからの付き合い．ビッグバンとブレイクの過程を楽しめた(？)世代．